Conseil national de recherches Canada
Division de génie électrique

National Research Council Canada
Division of Electrical Engineering

ERB-[illegible]

Étude préliminaire d'une base de données pour un poste de travail graphique 3-D

P. Chartier

avril 1987

87-04601/

Copyright 1987 par
Conseil national de recherches du Canada

Il est permis de citer de courts extraits et de reproduire des figures ou tableaux du présent rapport, à condition d'en identifier clairement la source.

Copyright 1987 by
National Research Council of Canada

Permission is granted to quote short excerpts and to reproduce figures and tables from this report, provided that the source of such material is fully acknowledged.

Des exemplaires supplémentaires gratuits peuvent être obtenus à l'adresse suivante:

Bureau des publications, Pièce 301
Division de génie électrique
Conseil national de recherches du Canada
Ottawa (Ontario) Canada
K1A 0R8

Additional copies are available free of charge from:

Editorial Office, Room 301
Division of Electrical Engineering
National Research Council of Canada
Ottawa, Ontario, Canada
K1A 0R8

ÉTUDE PRÉLIMINAIRE D'UNE BASE DE DONNÉES POUR UN POSTE DE TRAVAIL GRAPHIQUE 3-D

P. Chartier

ERB-999
CNRC N° 26862

AVRIL 1987

TABLE DES MATIÈRES

	Page
RÉSUMÉ/ABSTRACT	v
INTRODUCTION	1
ÉNONCÉ DU PROBLÈME	1
LES ÉLÉMENTS DE SOLUTION	2
APPLICATION AUX DONNÉES GRAPHIQUES	5
LES SOLUTIONS PROPOSÉES	6
SOLUTION PRATIQUE	6
SOLUTION IDÉALE	7
CONCLUSION	7
RÉFÉRENCES	7

LISTE DES FIGURES

1	Graphe d'une donnée numérique	3
2	Exemple d'une donnée numerique et de son decoupage sémantique	4

RÉSUMÉ

Le développement d'un capteur 3-D à haute vitesse entraîne des contraintes dans la gestion des données graphiques. Les méthodes de classification et de gestion applicables aux systèmes CAO/FAO ne sont pas utilisables parce qu'elles présument une variété de formes représentées avec un nombre limité de données. Les méthodes applicables pour les données météorologiques sont pertinentes pour des images comportant de grandes masses de données présentées de façon assez uniformes.

La caméra 3-D cause un problème composé tenant à la fois des problèmes météo, CAO/FAO, du traitement d'images et du traitement graphiques. Semblablement, la solution semble résider dans un croisement entre les méthodes spécifiques de gestions propres à ces domaines.

ABSTRACT

The development of a 3-D sensor carries a whole set of constraints in picture data management. Methods applicable to CAD/CAM are not practical because they assume a variety of formats with a limited number of data points. Weather data management methods do apply to images made of massive amounts of data of a somewhat uniform structure.

The 3-D scanner poses a problem combining the difficulties found in weather data manipulation, CAD/CAM, image processing, and graphics. Consequently, solution seems to reside in a fusion of the tools applicable to those fields.

ÉTUDE PRÉLIMINAIRE D'UNE BASE DE DONNÉES POUR UN POSTE DE TRAVAIL GRAPHIQUE 3-D

P. Chartier

INTRODUCTION

Le projet de recherche pour la caméra 3-D (1) a généré un projet de développement d'un poste de travail interactif permettant d'exploiter les images tridimensionnelles. Ce projet vise à combiner les capacités de la caméra 3-D à celles d'un éditeur graphique et d'un logiciel de commande de machine outil. Cette fusion représente une nouvelle approche au concept d'atelier flexible et se place à mi-chemin entre l'automatisation dite "dure" et la robotisation pure.

Un tel projet exige une gestion souple et efficace d'un volume phénoménal de données de toutes sortes et le présent rapport décrit les constatations préliminaires pertinentes à la réalisation de cette gestion.

ÉNONCÉ DU PROBLÈME

La caméra 3-D est un télémètre laser à balayage synchronisé capable de digitaliser des surfaces avec rapidité (0.6 s) et haute résolution en distance (10 bits) avec une définition moyenne (256 voxels × 256 voxels). D'autres modèles, conçus pour optimiser le coût ou la précision, ont aussi été mis au point. Les données générées par la caméra sont placées dans une mémoire tampon lors de l'acquisition et transférées à un mini-ordinateur pour faire le traitement. Ultimement, ces données vont se retrouver sur disque sous forme de fichier.

Il est aussi possible de prendre plusieurs vues d'un même objet de façon à le représenter sous toutes ses faces et de faire un collage des vues multiples. Les données brutes de la caméra doivent être corrigées pour les replacer dans un référentiel orthogonal. Ceci constitue la calibration.

Même avec plusieurs vues du même objet, il est possible que certaines concavités résistent à l'inspection. On peut altérer ces défauts, compléter des aires manquantes à l'aide de l'éditeur de surfaces.

Dès qu'un objet a été modifié de façon satisfaisante, on peut générer un fichier de commandes pour machine outil qui servira à reproduire cet objet.

On peut aussi envisager l'utilisation des données pour faire de l'inspection de défauts et éventuellement comme support pour un véritable système de vision. Nous avons réussi à atteindre le stade où nous pouvons reproduire une vue d'un objet et nous avons produit une copie solide de cette vue. Il ne faut pas confondre avec le terme "Hard-copy"; nous avons reproduit un objet en trois dimensions.

Toutes ces étapes, même dans le cas d'une seule vue, sont susceptibles de générer un fichier de coordonnées comportant soit 128 K-octets (pour les coordonnées implicites) ou 384 K-octets (pour les tableaux X, Y, Z dès l'étape de calibration). Il faut aussi souligner que le fichier de commande de la machine-outil à contrôle numérique peut être beaucoup plus important.

Dans un environnement de production, on aura affaire à plusieurs objets ayant chacun leur niveau d'achèvement. Leurs données vitales seront nécessairement distinctes. Nous devons donc disposer d'un système cohérent de gestion qui puisse cohabiter avec le système de traitement sans monopoliser les ressources de la machine.

LES ÉLÉMENTS DE SOLUTION

Aucun système de bases de données graphiques n'existe sur le marché. Les systèmes de base de données relationnelles, entre autres, offrent une séparation entre le logiciel d'exploitation des données et la représentation des données elles-mêmes. Cette caractéristique est désirable. Par contre, les modèles courants reposent sur un axiome qui veut que la structure de la base de données soit représentable en terme des données qu'elle contient. La figure 1 représente une donnée typique par des relations entre ses divers attributs. En d'autres termes, un objet est représenté par des opérations sur des champs de données et l'ensemble (opérations, données) est représentable dans des champs de données.

Il est donc normal de disposer, dans un système conventionnel, de tableaux représentant l'état de la base de données et d'effectuer des opérations de la même façon sur les données internes du système que sur les données d'un usager. On peut établir quatre classes sémantiques de données et cette classification, sans se prétendre universelle, éclaire un peu le problème.

Il y a d'abord les valeurs proprement dites. Ces données sont scalaires et réfèrent à des unités de mesure ou à des réalités concrètes. On a par exemple des montants d'argent, des salaires, des mensurations qui sont des valeurs numériques. On a aussi des ratios, des pourcentages, des

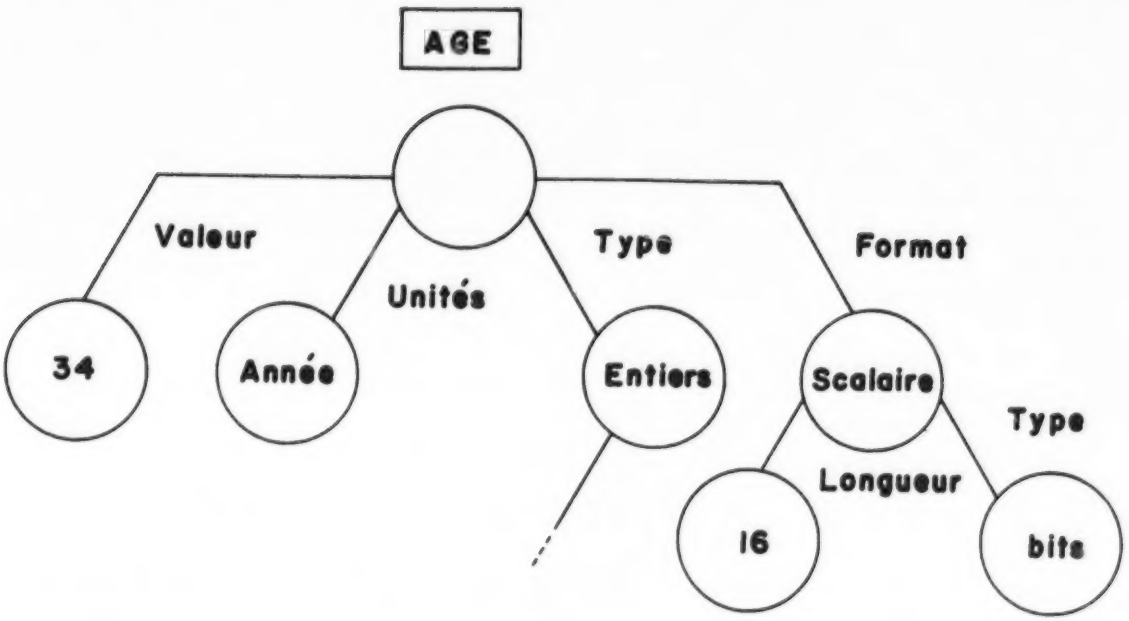

Figure 1. Graphe d'une donnée numérique.

taux qui sont des nombres purs établissant des liens avec d'autres données. Enfin, on a des valeurs symboliques que sont les noms, les adresses, les numéros de série ou les codes d'inventaire. Cette classe sera désignée par le préfixe "iso", du grec ἴσος, égal. Ces données servent à modéliser une réalité qu'on veut représenter dans la base de données.

Les isodonnées vont être représentées dans un cadre quelconque dans la base de données. Ce cadre lui-même, soit un scalaire, un tableau, etc., sera représenté par un ou plusieurs paramètres. Ces paramètres ont une existence propre en ce qu'un jeu de paramètres peut décrire un groupe d'isodonnées quelconque et qu'on peut effectuer sur eux des opérations sans se soucier, a priori, des isodonnées qu'ils servent à structurer. Cette classe sera la classe épi d'une racine grecque signifiant "sur". On aura, dans cette classe, des dimensions de tableaux, des tables de pointeurs, des index.

Le type d'une donnée quelconque, que ce type soit atomique ou composé, sera appelé information "péri". C'est la classe des informations décrivant les unités de calcul, tels les "centimètres", "degrés" ou autres et les types d'organisation, tels les "tableaux", "index", "vecteurs". Les péri-données régissent les conversions, les méthodes d'accès et peuvent servir à valider la sémantique des opérations demandées.

Une quatrième classe sert à recevoir toute donnée qui sert à la gestion des autres classes et du système en général. C'est la classe META. On y retrouve des informations essentiellement symboliques dont la valeur sera directement ou indirectement précisée par une des trois autres classes. Par exemple, pour un fichier de nombres réels, le nom de ce fichier sera une information META, mais la valeur de ce nom sera du type ISO. Pour ce même fichier, la longueur du fichier sera de la classe EPI et le type RÉEL sera de la classe PERI. La longueur du nom du fichier sera de classe EPI elle aussi et le type de données servant à coder ce nom sera de classe PERI.

Avec ces quatre classes, soit META, PERI, EPI et ISO, on peut élaborer des règles d'utilisation.

(a) Une isodonnée modélise la réalité et ne réfère à aucune autre donnée.

(b) Une épidonnée peut référencer une isodonnée ou une autre épidonnée.

(c) Une péridonnée peut référencer une isodonnée une épidonnée ou une autre péridonnée.

(d) Une métadonnée peut référencer n'importe quelle autre classe.

Par référencer, nous signifions que la valeur résultant de l'application d'une fonction à une classe K_a appartient à une classe K_b.

$$\left(F(K_a) \rightarrow K_b\right) : \left((K_a = K_b) \vee (K_a \neq K_b)\right)$$

Il faut aussi retenir la distinction entre un objet ou symbole et sa valeur. Par exemple, le nom de notre fichier est une métadonnée, i.e., un symbole. La valeur de ce nom, par contre est du type ISO. Un type composé sera de classe PERI mais sa valeur sera un ensemble de données PERI, EPI et ISO. Ces différents niveaux sont illustrés dans la figure 2.

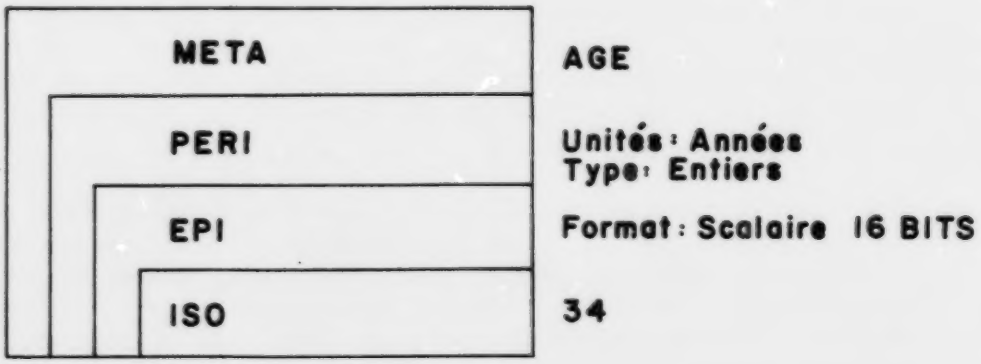

Figure 2. Exemple d'une donnée numerique et de son decoupage sémantique.

La fonction de référence liant une donnée à une autre peut recevoir des arguments de classe quelconque et la classe du résultat sera lui aussi d'une classe quelconque. On donnera aux fonctions une classe qui correspond à celle du résultat.

Dans un système de bases de données conventionnel, on retrouve deux classes de données. Les données ISO et les données EPI correspondent aux champs et à leur contenu. Tout l'accès aux données PERI ou META est relégué à la procédure de définition de la base de données. Dans le cas du traitement graphique, on a, de plus, une situation où le volume d'iso-données déborde la capacité des champs permis.

Pour terminer, les systèmes de base de données limitent la classe des fonctions au niveau ISO. On ne peut donc interroger une base de données que pour en ressortir des valeurs.

Que se passe-t-il si une fonction META peut recevoir un argument META (qui peut être lui-même le résultat d'une fonction)? On aura une fonction qui reçoit un symbole et qui produit un autre symbole.

Notre système de base de données ressemblera étrangement à un langage ou des symboles sont utilisés pour en manipuler d'autres. On pourra alors décrire un objet en termes de prédicats et le rôle de la base de données sera de vérifier ou d'infirmer ces prédicats.

APPLICATION AUX DONNÉES GRAPHIQUES

Un aspect spécifique des données graphiques tient de la variété des représentations possibles, incompatibles les unes avec les autres, propres à représenter un objet.

Dans notre système, on distingue six classes.

(1) Les données brutes qui sont des vecteurs de distances non calibrées.
(2) Les données calibrées, tableaux de coordonnées spatiales x, y et z.
(3) Les modèles "fils de fer", tableaux décrivant la connectivité entre les points de la surface de l'objet.
(4) Les modèles comprimés, où l'image est représentée par des fonctions générant des splines, des hologrammes, des textures, etc.
(5) Les projections où l'objet est représenté en deux dimensions pour fins d'affichage.
(6) Les programmes APT où l'objet est représenté comme une séquence d'opérations de machinage.

Chacune se prête avec plus ou moins de bonheur à divers types de traitement. De plus, le passage d'une classe à une autre n'est pas toujours reversible.

La solution simpliste consistant à emmagasiner un objet sous toutes ses représentations possibles n'est pas généralisable non plus, puisqu'on doit ajouter aux représentations elles-mêmes des informations supplémentaires modélisant la correspondance entre les diverses représentations. Le volume total, dans notre cas, semblait aberrant.

L'autre solution possible est de représenter un objet d'une façon telle que les autres représentations puissent être générées rapidement sur demande. Cette solution est séduisante mais peu pratique puisque là, ce sont les temps d'accès et de calcul qui deviennent aberrants. Cette représentation, reposant sur un ensemble de générateurs, est dite symbolique.

LES SOLUTIONS PROPOSÉES

La nature du projet demandant une solution pratique à court terme nous amène à proposer une solution en deux volets. Un des ces volets est utilisable à court terme mais est très déficient. L'autre, plus complet, n'est pas réalisable à court terme (6 mois).

SOLUTION PRATIQUE

Pour l'immédiat, il importe d'accommoder des logiciels expérimentaux fonctionnant avec une variété de fichiers en limitant les coûts d'installation. Pour ce faire, nous sommes à évaluer un système de base de données commerciales où seront conservées et mises à jour les données META, PERI et une sous-classe EPI. Les seules fonctions possibles seront limitées aux fonctions ISO des données de la base de données.

Les programmes d'application, eux, consulteront la base de données pour trouver des informations qu'ils pourront utiliser pour accéder les données ISO des images à traiter. Ces dernières se trouveront dans des fichiers identiques à ceux en usage actuellement et leurs noms seront conservés dans la base de données.

Un inventaire des programmes sera tenu dans le système de gestion de bases de données et permettra l'entretien et servira de base à un système de menu interactif qui pourra être développé par la suite.

SOLUTION IDÉALE

La solution idéale demande un langage de programmation graphique. Les objets résidant dans la base de données seront représentés par des "programmes".

Cette solution se rapproche de la solution symbolique mais pour limiter le temps d'accès, ces programmes pourront être de simples accès à des données, ce qui nous rapproche de la solution "simpliste". Une façon d'éviter l'explosion du volume des données sera d'associer tout objet à une classe et de ne représenter que les différences entre un objet et sa classe à l'aide d'un mécanisme d'hérédité. Cette solution devra faire appel à du matériel spécialisé propre à accélérer le traitement des représentations symboliques.

Il existe des publications décrivant une approche semblable (2). Par contre, des demandes de renseignements supplémentaires adressées aux auteurs sont toujours sans réponse.

CONCLUSION

La gestion des données graphiques peut se présenter sous deux aspects. Une approche "gestion pure" où le système de bases de données permet un suivi d'un objet dans le système et une approche "opérationelle" où le système modélise réellement les données. La première approche, bien que limitée, nous permettra un peu plus de rigueur dans l'utilisation des images de la caméra 3-D.

La seconde, à l'état de projet permettra de jeter les bases d'un véritable système intégré capable de simuler un système de vision grâce à un traitement symbolique des images.

RÉFÉRENCES

1. M. Rioux et L. Cournoyer. Travaux préliminaires à la mise au point d'un capteur de vision tridimensionnelle. ERB-944, Conseil national de recherches Canada, Ottawa (Ont.). CNRC 20189.

2. A.P. Buchmann et J.M. Gerzso. Handling herogenously formatted data in an object oriented database environment. Computer Graphics '85 Conference Proceedings, Technical Sessions, Vol. III. ISBN 0-941514-09-9 (Vol. III). The National Computer Graphics Association 1985. p. 645.